Exercise work

Name: Pekka Lehtola

Do you need help/comments:

-

Topic:

Topic for exercise work is two player game called Indians vs. Cowboys.

Schedule:

- Week 1: Core game mechanics working.
- Week 2: UI
- Week 3: Random events.
- Final: Updated graphics.

How I achieve goals:

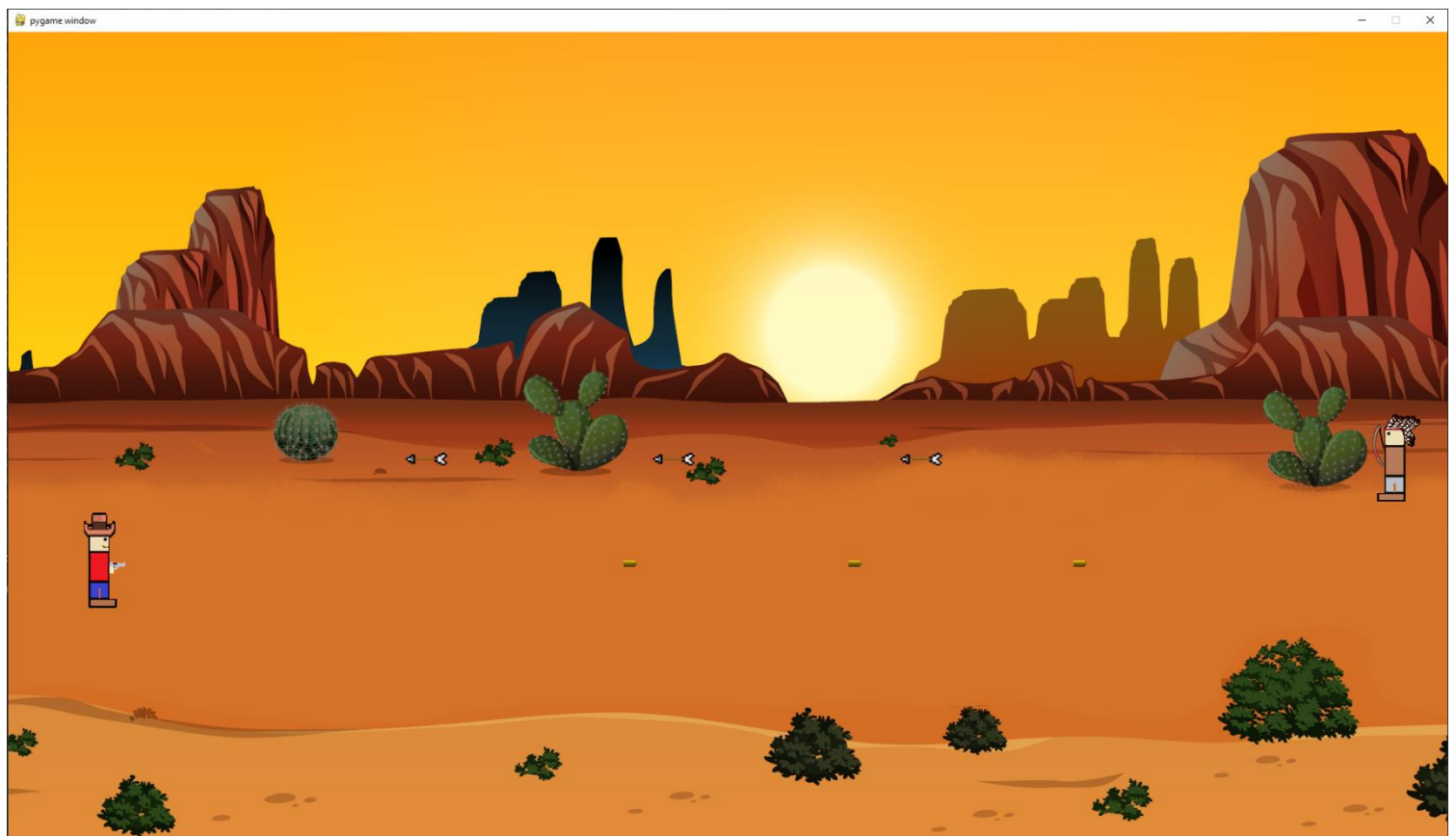Working hard on the project and scour the internet about pygame library.

Weekly exercises:

I have done 100% of the weekly exercises.

Time spent on weekly exercises 8h/week.

Project work so far:

Players can move, reload, shoot multiple projectiles at once and characters can be hit.

Screen capture of code:

Main:

```python
# File name: main
# Author: Pekka Lehtola
# Description: main file exercise work

import ...

# Initialize pygame
pygame.init()


# Sets up screen size
screen = pygame.display.set_mode([SCREEN_SIZE_HOR, SCREEN_SIZE_VER])

running = True

# Indians initial location
indian.rect.x = 1820
indian.rect.y = 640

# Cowboys initial location
cowboy.rect.x = 100
cowboy.rect.y = 640

while running:

    # Detects events while game is running.
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Pygame detects what keys are pressed.
    pressed_keys = pygame.key.get_pressed()

    # The term used for rendering objects is blitting.
    # Surf : Surface, Rect : Rectangle.
    # Rectangle is used for collide detection
    # Surface can be image or color.

    for characters in all_sprites:
        screen.blit(characters.surf, characters.rect)

    # Reload function
    indian.reload(pressed_keys)
    cowboy.reload(pressed_keys)

    # Shooting function
    indian.shoot(pressed_keys)
    cowboy.shoot(pressed_keys)

    # Moving function
    indian.move(pressed_keys)
    cowboy.move(pressed_keys)

    # Bullet travel and hit detection
    for ammo in indian.shot_ammo:

        ammo.ammo_shot(indian, cowboy)
        screen.blit(ammo.surf, ammo.rect)

    for ammo in cowboy.shot_ammo:

        ammo.ammo_shot(cowboy, indian)
        screen.blit(ammo.surf, ammo.rect)


    # Update screen
    pygame.display.flip()

    # Background update
    screen.blit(BG, (0,0))

    # Sets Frame duration
    time.sleep(FRAME_DURATION)

pygame.quit()
```

Character 1/2:

```python
# File name: character
# Author: Pekka Lehtola
# Description: Character class that is derived from pygame sprites

import pygame
from config import *
from ammo import *

class Character(pygame.sprite.Sprite):

    def __init__(self, sprite, name):
        super(Character, self).__init__()
        self.name = name
        self.surf = pygame.image.load(sprite)                            # Surface image
        self.surf.set_colorkey(GREEN_SCREEN)                             # Defining see through color
        self.rect = self.surf.get_rect(size=(40,114), center=(0, 0))    # Sets up characters hitbox
        self.ammo = []                                                   # Reloaded ammo
        self.shot_ammo = []                                             # Ammo that character has shot
        self.reloaded = True
        self.shooting_speed = SHOOTING_SPEED
        self.reload_speed = RELOAD_SPEED

    # Reloading function
    def reload(self, pressed_keys):

        # Detects which player is reloading
        if self.name == "Indian":

            # Reload speed updates each frame creating time delay how fast ammo is loaded.
            if self.reload_speed < RELOAD_SPEED:

                self.reload_speed += 1

                if self.reload_speed == RELOAD_SPEED:
                    self.reloaded = True

            # Restarts reload speed and appends ammo to magazine.
            elif self.reloaded == True and pressed_keys[K_k]:

                self.reload_speed = 0

                self.ammo = []

                for ammo in range(0, 3):

                    ammo = Ammo("images/arrow.png", "arrow")    # Define which ammo is inserted.

                    self.ammo.append(ammo)

                self.reloaded = False

        # Cowboys reload function is indentical
        if self.name == "Cowboy":...
```

```python
 76         # Shooting function
 77         # Work similary to reload function with time delay.
 78         def shoot(self, pressed_keys):
 79
 80             if self.name == "Cowboy":
 81
 82                 if self.shooting_speed == SHOOTING_SPEED and pressed_keys[K_SPACE]:
 83
 84                     self.shooting_speed = 0
 85
 86                     if self.reloaded:
 87
 88                         try:
 89                             ammo = self.ammo.pop(-1)        # Removes last ammo object from ammo list
 90                             self.shot_ammo.append(ammo)     # Adds that ammo to shot ammo list
 91                             ammo.rect.x = self.rect.x + 30  # Ammos intial location is players X
 92                             ammo.rect.y = self.rect.y + 63  # And Y coordinates
 93                             ammo.shot = True
 94
 95                         except:
 96
 97                             pass
 98
 99                 if self.shooting_speed < SHOOTING_SPEED:
100
101                     self.shooting_speed += 1
102
103             # Indians reload function is identical
104             if self.name == "Indian":...
124
125         # Moving function
126         def move(self, pressed_keys):
127
128             # Detects keyboard inputs and calculates new location for character.
129             if self.name == "Indian":
130
131                 if pressed_keys[K_UP]:
132                     self.rect.move_ip(0, -MOVEMENT_SPEED)
133                 if pressed_keys[K_DOWN]:
134                     self.rect.move_ip(0, MOVEMENT_SPEED)
135                 if pressed_keys[K_LEFT]:
136                     self.rect.move_ip(-MOVEMENT_SPEED, 0)
137                 if pressed_keys[K_RIGHT]:
138                     self.rect.move_ip(MOVEMENT_SPEED, 0)
139
140             if self.name == "Cowboy":
141
142                 if pressed_keys[K_w]:
143                     self.rect.move_ip(0, -MOVEMENT_SPEED)
144                 if pressed_keys[K_s]:
145                     self.rect.move_ip(0, MOVEMENT_SPEED)
146                 if pressed_keys[K_a]:
147                     self.rect.move_ip(-MOVEMENT_SPEED, 0)
148                 if pressed_keys[K_d]:
149                     self.rect.move_ip(MOVEMENT_SPEED, 0)
150
151             # Detects if character has reached levels edges and stops them moving off the screen.
152             if self.rect.left < 0:
153                 self.rect.left = 0
154             if self.rect.right > SCREEN_SIZE_HOR:
155                 self.rect.right = SCREEN_SIZE_HOR
156             if self.rect.top <= 380:
157                 self.rect.top = 380
158             if self.rect.bottom >= SCREEN_SIZE_VER:
159                 self.rect.bottom = SCREEN_SIZE_VER
160
161     #def scale(self):
162
163 # Creates indian and cowboy objects.
164 # Also define graphics to them.
165 indian = Character("images/indian_1.png", "Indian")
166 cowboy = Character("images/cowboy_1.png", "Cowboy")
167
168 # Adds indian and cowboy to sprite group.
169 all_sprites = pygame.sprite.Group()
170 all_sprites.add(indian)
171 all_sprites.add(cowboy)
```
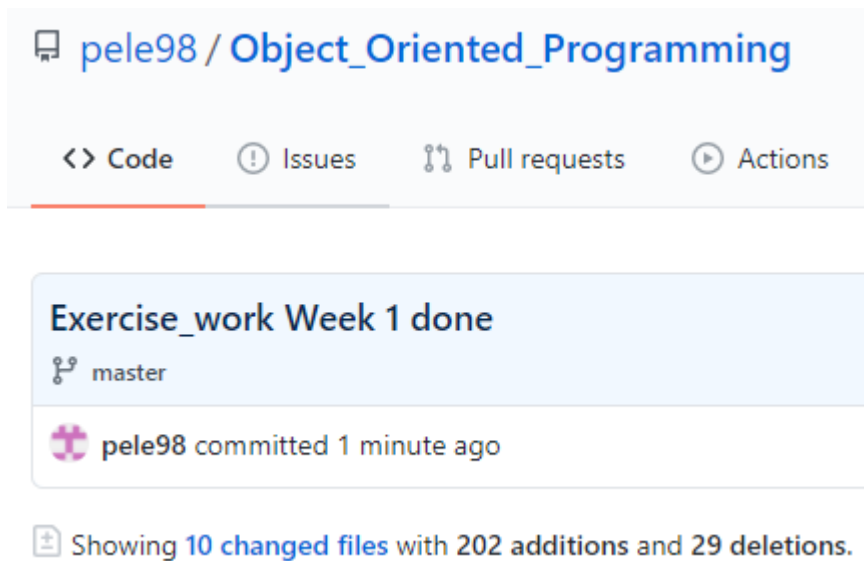
Ammo:

```python
# File name: character
# Author: Pekka Lehtola
# Description: Ammo class for characters to shoot.

import pygame
from config import *

# Ammo is derived from pygame sprites
class Ammo(pygame.sprite.Sprite):

    def __init__(self, ammo, name):
        super(Ammo, self).__init__()
        self.name = name
        self.surf = pygame.image.load(ammo)      # Image
        self.surf.set_colorkey(GREEN_SCREEN)     # See through color
        self.rect = self.surf.get_rect()         # Hitbox

        self.shot = False

    # Function for shot ammo
    def ammo_shot(self, shooter, enemy):

        if self.shot == True:

            # If the shooter is indian arrow travels to left
            # If arrow hits cowboy, cowboy is killed.
            if shooter.name == "Indian":
                self.rect.move_ip(-BULLET_VELOCITY, 0)

                if self.rect.colliderect(enemy.rect):
                    enemy.kill()

            # Same as indian but bullet travels right
            if shooter.name == "Cowboy":
                self.rect.move_ip(BULLET_VELOCITY, 0)

                if self.rect.colliderect(enemy.rect):
                    enemy.kill()
```

Config:

```python
# File name: config
# Author: Pekka Lehtola
# Description: Config file for constant variables.

import pygame

FRAME_DURATION = 1/120

SCREEN_SIZE_VER = 1080
SCREEN_SIZE_HOR = 1920

BULLET_VELOCITY = 30
MOVEMENT_SPEED = 10

SHOOTING_SPEED = 10
RELOAD_SPEED = 5

WHITE = (255, 255, 255)
BLUE = (0, 0, 255)
BLACK = (0, 0, 0)
```

```python
# Green screen color is used to remove background from sprites.
GREEN_SCREEN = (0, 177, 64)

BG = pygame.image.load("images/desert.png")

# Keyboard inputs are located here.
from pygame.locals import (
    RLEACCEL,
    K_UP,
    K_DOWN,
    K_LEFT,
    K_RIGHT,
    K_ESCAPE,
    K_w,
    K_s,
    K_d,
    K_a,
    K_SPACE,
    K_j,
    K_k,
    K_r,
    KEYDOWN,
    QUIT,
)
```

pele98 / Object_Oriented_Programming

<> Code   ⊙ Issues   ⇅ Pull requests   ▷ Actions

Exercise_work Week 1 done

⑂ master

🔴 pele98 committed 1 minute ago

⊞ Showing **10 changed files** with **202 additions** and **29 deletions**.

Work done this week: 8h

Self-assessment:

This exercise was easy/difficult/ok/etc. for me because…

Ok

Doing this exercise, I learned…

Oppisin käyttämään pygame kirjastoa hieman enemmän.

I am still wondering…

Pygame eventit eivät oikein vielä onnistuneet.

I understood/did not understand that… ; I did/did not know that… ; I did/did not manage to do…

-